Network Intruder Location Using Markov Decision Processes *

Extended Abstract for RAID 2000: Third International Workshop on Recent Advances in Intrusion Detection

Topic Category: Innovative Approaches/New IDS Methodologies and Technologies

T. Darling and M. A. Shayman
Department of Electrical and Computer Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742
Email: shayman@eng.umd.edu

April 26, 2000

1 Intrusion and Misuse Location

In recent years, there has been considerable progress in developing systems for the *detection* of network intrusion and misuse. In contrast to the large amount of work on intrusion/misuse detection, there has been much less research reported on the crucial related problem of locating the source(s) of an attack once it is detected. Because of *IP spoofing*, the source address in an attack packet cannot be relied upon to disclose the true source of the attack.

Recently, researchers from North Carolina State University and MCNC have proposed DECIDUOUS, a security management framework for identifying the sources of network-based intrusions [2]. DECIDUOUS is constructed on top of IETF's IPSEC/ISAKMP infrastructure. For attack source identification in a single administrative domain, it has the advantage of not requiring any new network protocols.

The key idea underlying DECIDUOUS is that of *dynamic security associations* (SA). Obviously one way to prevent spoofing and ensure identifiability of attack sources is to establish SAs between every pair of nodes that might ever need to communicate. However, this may be too expensive from a computational viewpoint. IPSEC processing overhead would occur even when there are no attacks. Static SAs between a limited collection of pairs of communicating nodes would not guarantee attacker identification, especially with the increasing sophistication of attackers who share vast databases of network vulnerability information [3]. Widespread static SAs may also be unacceptable from an

*This research was supported in part by NSF under Grant ECS-9626399 and by the NSA Laboratory for Telecommunications Science under contract MDA90499C2521

administrative or policy viewpoint. The solution proposed by DECIDUOUS is to construct a security management module that dynamically decides when and where to establish IPSEC SAs.

For dynamic SAs to be effective at locating intruders, it is critical to have an efficient, if not optimal, decision algorithm to determine when and where to establish SAs. We will show that this problem can be formulated as a Markov decision process (MDP). This opens up the possibility of using techniques for the solution (and approximate solution) of MDPs to solve the sequential decision problem of SA placement. In fact, the problem is an example of an important class of MDPs that we refer to below as *hierarchical diagnosis problems.*

2 The Hierarchical Diagnosis Problem

We define the general *hierarchical diagnosis problem* as follows: Let S be the collection of all components. For component i, let X_i be the indicator function for the component being faulty. For each subset $A \subset S$, it is "possible" to perform a test to determine if A contains a faulty component. The cost of such a test is represented by a random variable C_A. In reality, the structure of the system and the properties of available diagnostic tools will make testing impossible for certain subsets A. This can be modeled by setting $C_A = \infty$. Finally, a joint distribution for the random variables $\{X_i\}$ is given. In the special cases of mutually exclusive or independent faults, it suffices to specify the probability p_i that component i is faulty for each i. Given this model, the problem is to determine the optimal order of subsets to be tested to locate the faulty component(s).

In the standard sequential diagnosis problem, only individual components may be tested. This corresponds to the special case of the hierarchical diagnosis problem in which $C_A = \infty$ whenever $|A| > 1$.

We focus on the important special case of the hierarchical diagnosis problem in which the subsets that may be tested are constrained by a directed tree. A directed spanning tree for S is given. For each $i \in S$, let A_i denote the subset consisting of i together with all its descendents. Only the subsets A_i may be tested; i.e., the testing cost for every other subset is infinite. Thus, there is a test associated with each node of the tree. A test corresponding to a node i will be positive if node i or any of its descendents are faulty.

We restrict our attention to the case where there is exactly one faulty component. In the literature, this is referred to as the *mutually exclusive fault* (MEF) problem. Suppose that there is a probability p_i that component i is faulty. For a subset A_i of S, let $p(A_i) = \sum_{j \in A_i} p_j$. If subset A_i is tested, the result will be positive with probability $p(A_i)$ and negative with probability $1 - p(A_i)$. If the result is positive, the fault is isolated to the subtree consisting of the nodes in A_i; if negative, it is isolated to the subtree consisting of the nodes in $S - A_i$.

More generally, suppose that a sequence of tests has been performed that has isolated the fault to a subtree S' of the initial tree S. If the subset A_i is tested, the fault will be isolated to either $A_i \cap S'$ or $S' - A_i \cap S'$ depending on whether the result is positive or negative, respectively. The probability of the former is $p(A_i \cap S')/p(S')$, while the probability of the latter is $1 - p(A_i \cap S')/p(S')$. Thus, we have a Markov decision process in which the state space consists of all subtrees of the initial tree S. There is an admissible action in state S' for each node $i \in S'$. If action i is taken, there are two possible successor states, namely $A_i \cap S'$ and $S' - A_i \cap S'$. The cost of taking action i in state S' is C_{A_i}. The MDP is a *stochastic shortest path problem* in which the terminal states consist of those subtrees S' that have $|S'| = 1$–i.e., single-node subtrees.

3 Intruder Location as an MDP

We now show that the problem of using dynamic security associations to locate intruders/misusers in an IP-based network can be formulated as a tree-hierarchical diagnosis problem. Hence optimal policies for establishing dynamic SAs can be obtained by using techniques available for solving the type of MDPs introduced above.

As in the DECIDUOUS project [2], we assume that an intrusion detection system is in place that can detect attack packets that arrive at a host or router. If a security association is established between node i (viewed as sender) and node j (viewed as receiver), then if an attack packet is detected at j it can be determined whether it was forwarded by i. We make the assumption that shortest path routing is used; e.g., consider an autonomous system using OSPF in the Internet. This means that there is a spanning tree rooted at j that is used to route all packets with final destination j. When a packet destined for j reaches an intermediate node i, i forwards it to its parent node in the spanning tree. Suppose that a security association is established between i and j. If an attack packet is then received at j, it can be determined whether this packet was forwarded by i. If the answer is positive, then the attack source is localized to the subtree consisting of i together with all of its descendents. If the answer is negative, then the attack source is localized to the subtree consisting of all nodes other than i and its descendents. (This assumes a single source attack.) Consequently, the problem of determining the optimal sequence of SAs is equivalent to the hierarchical diagnosis problem described above and corresponds to an MDP. Note that if the attacker is not continuously sending attack packets, then each time we establish a new SA we will need to wait a random length of time until the next attack packet is received.

In this formulation, the probability p_i is the probability that the attacker is located at node i. These probabilities may be initially assigned based on prior assessment of network vulnerabilities and may be updated using whatever information regarding attack source location is available from the system that detects the attack. The single-stage cost function $C_i := C_{A_i}$ is the "cost" of establishing an SA between node i and the victim node.

3.1 Features and Neurodynamic Programming

In general, the optimal stationary policy for the stochastic shortest path problem for the tree-hierarchical MEF diagnosis problem will depend on the topological structure of the tree, the fault probabilities p_i, and the single-stage cost function C_i. For the remainder of this paper, we consider the special case where the probabilities and the single-stage cost function are constant. In other words, we assume that all nodes are equally likely as the attack source and the establishment of each security association incurs a cost which is independent of the node. Under these assumptions it is clear that the policy is determined by the graph topology.

As is pointed out in [2], the choice of policy is obvious in the case of a *linear* topology. In this case, an optimal policy (which is also worst-case optimal) corresponds to choosing the security association with a middle node–i.e., such that the difference in the number of nodes in the two successor subtrees is at most one. In this case, the number of SAs required to locate the attacker is deterministic and is a logarithmic function of the number of nodes. At the opposite end of the spectrum is the case where all nodes are directly connected to the victim node. In this case, each subset A_i consists of a single node and the optimal cost depends linearly on the number of nodes. However, in the case of general topology, the solution is not at all obvious. Furthermore, exact solution by dynamic programming becomes impractical for more than about 20 nodes. Consequently, it is natural to consider the use of one of the variants of neurodynamic programming [1] [4].

We are in the process of exploring the use of value-function approximation using a linear combination of features. A feature function assigns a number to each state–i.e., to each tree. Possible choices for features include number of nodes, average path length (from root to leaf node), longest path (from root to leaf node), number of leaf nodes, average number of children per node, maximum number of children per node, etc. Since the approximation architecture is linear, nonlinear functions of features, such as the product of two features, must be treated as additional potential features. At this point we have done extensive computations that indicate that the product of the average number of children per node and the maximum number of children per node together with the number of nodes gives a pair of features that seem to yield good approximations for the optimal value function. Additional results are forthcoming.

References

[1] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

[2] H. Y. Chang, R. Narayan, S. F. Wu, B. M. Vetter, X. Wang, M. Brown, J. J. Yuill, C. Sargor, F. Jou, and F. Gong. Deciduous: Decentralized source identification for network-based intrusions. In *Proceedings of International Symposium on Integrated Network Management*, pages 701–714, Boston, MA, May 1999.

[3] T. Longstaff. Panel: Preparing for internet threats in the years 2001 to 2010. In *National Information Systems Security Conference*, Arlington, VA, October 1999.

[4] R. Sutton and A. Barto. *Reinforcement Learning.* MIT Press, Cambridge, MA, 1998.